

# Information-Based Guidance and Control Architecture for Multi-Spacecraft On-Orbit Inspection

Yashwanth Kumar Nakka<sup>\*</sup>, Wolfgang Hönig<sup>†</sup>, Changrak Choi<sup>‡</sup>, Alexei Harvard<sup>§</sup>, Amir Rahmani<sup>¶</sup>, and Soon-Jo Chung<sup>||</sup>

We present an architecture for inspection or mapping of a target spacecraft, referred to as chief, in an orbit around Earth using multiple spacecraft, referred to as deputies (or) observers, in stable Passive Relative Orbits (PROs). We use an information gain approach to directly consider the trade-off between gathered data and fuel/energy cost. The four components of our architecture are: 1) information estimation, 2) state estimation, 3) motion planning for relative orbit initialization and reconfiguration, and 4) relative orbit control. The information estimation quantifies the information gain during inspection of a spacecraft, given past and potential future poses of all spacecraft. The estimated information gain is a crucial input to the motion planner, which computes PROs and reconfiguration strategies for each of the observers to maximize the information gain from distributed observations of the target spacecraft. The resulting motion trajectories jointly consider observational coverage of the target spacecraft and fuel/energy cost. For the PRO trajectories, we design a fuel optimal attitude trajectory that minimizes rest-to-rest energy for each observer to inspect the target spacecraft. We validate our architecture in a mission simulation to visually inspect the target spacecraft.

## Nomenclature

LVLH	= Local-Vertical Local-Horizontal Frame
ECI	= Earth Centered Inertial Frame
PRO	= Passive Relative Orbit
POI	= Points Of Interest on surface of the target spacecraft
FoV	= sensor field of view
$N$	= number of deputies (or) observers
$(x, y, z)$	= relative orbit coordinates in LVLH frame
$\mathbf{x}_i$	= state of the $i^{\text{th}}$ observer spacecraft
$\bar{\mathbf{x}}_i$	= nominal trajectory of the $i^{\text{th}}$ observer spacecraft
$m$	= mass of the observer spacecraft
$n$	= mean motion of the target spacecraft
$\mathbf{u}$	= control input for relative orbit dynamics
$\mathcal{U}$	= convex control constraint set
$\mathbf{A}$	= state transition matrix for continuous time
$\mathbf{B}$	= input matrix for continuous time
$\mathbf{A}_d$	= state transition matrix for discrete time
$\mathbf{B}_d$	= input matrix for discrete time
$T$	= time step
$\mathbf{q}$	= attitude of the observer spacecraft as quaternion
$\bar{\mathbf{q}}$	= nominal attitude trajectory of the observer spacecraft
$\boldsymbol{\omega}$	= angular velocity
$\mathbf{J}$	= inertia matrix of the observer spacecraft
$\boldsymbol{\tau}$	= torque

<sup>\*</sup>PhD Candidate, GALCIT, California Institute of Technology, Pasadena, CA, 91125, USA

<sup>†</sup>Postdoctoral Researcher, GALCIT, California Institute of Technology, Pasadena, CA, 91125, USA

<sup>‡</sup>Robotics Technologist, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91104, USA

<sup>§</sup>Research Engineer, GALCIT, California Institute of Technology, Pasadena, CA, 91125, USA

<sup>¶</sup>Robotics Systems Engineer, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91104, USA

<sup>||</sup>Bren Professor of Aerospace; Jet Propulsion Laboratory Research Scientist, California Institute of Technology, Pasadena, CA, 91125, USA

$\mathbf{A}_{1q}, \mathbf{A}_{1w}, \mathbf{A}_{2w}$	=	state transition matrices of linearized attitude dynamics
$H$	=	information cost
$\mathbf{s}$	=	sampled POI on the target spacecraft
$\mathbf{p}$	=	pose of sensor on the observer spacecraft
$\mathcal{P}$	=	set of all sensor poses
$w$	=	variance from prior model
$f(\mathbf{p}, \mathbf{s})$	=	variance of estimating POI $\mathbf{s}$ with a sensor at pose $\mathbf{p}$
$\phi(\mathbf{s})$	=	relative importance of POI $\mathbf{s}$
$d_{min}$	=	minimum safe distance from the observer to the target spacecraft
$d_s$	=	desired sensing distance from the observer to the target spacecraft
$\omega_{low}, \omega_{mid}, \omega_{high}$	=	loop frequency of low-, mid-, and high-rate components

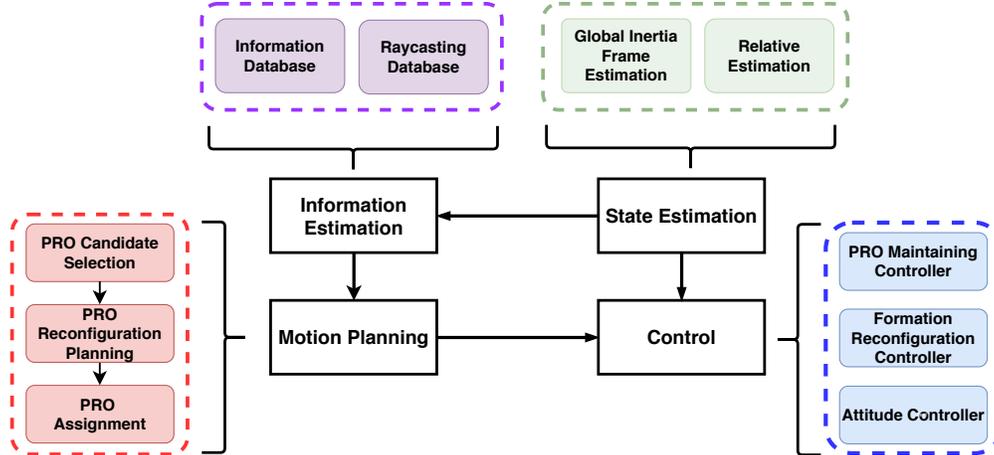
## I. Introduction

Autonomous on-orbit servicing [1] and inspection of a spacecraft is a key enabling technology for increasing the mission lifetime, improving the safety of the human spaceflight, and gaining an insight into the health of the spacecraft. Inspection and mapping of known or unknown space objects in Earth’s orbit is the first essential step towards on-orbit servicing capabilities that include refueling, repairing, assembling, and upgrading of space assets [2, 3]. State-of-the-art approaches for on-orbit inspection include robotic manipulators with rails for extended work space, single servicing spacecraft [4, 5], and astronaut intervention. These approaches have severe limitations: the capability of manipulators are limited due to mechanical constraints [4]; the single service orbiter has payload constraints; and astronaut intervention is expensive, and potentially dangerous. A team of collaborating spacecraft has the potential to provide augmented capabilities with enhanced robustness and versatility at a reduced cost and time [6, 7]. Thus, we propose to use a distributed spacecraft formation for on-orbit inspection. The multi-spacecraft approach has additional challenges [8, 9] of coordination, synchronization, and communication. The technologies for small spacecraft such as CubeSats are maturing quickly, making multi-spacecraft servicing closer to a reality [10–13]. In this work, we present a Guidance, Navigation and Control (GNC) architecture for inspection or mapping of a target spacecraft, referred to as chief, in an orbit around Earth using multiple spacecraft, referred to as deputies, in stable Passive Relative Orbits (PROs).

We assume that the target spacecraft is in a low Earth orbit, and the deputies are deployed from the target spacecraft. We design stable relative parking orbits for initial formation deployment. From the onset of formation deployment, our guidance algorithm focuses on designing PROs and reconfiguration strategies for each of the deputies to maximize the information gain from distributed observation of the target. The resulting motion trajectories jointly consider (a) observational coverage of the target spacecraft, (b) fuel/energy cost and, (c) collision checking while iteratively acting upon updates from estimation. The planner strives towards computational efficiency to make real-time on-board computation possible, while making trade-off between optimality as needed. The capabilities of the proposed planner framework are demonstrated on a design reference mission involving up to 5 spacecraft performing coordinated inspection of a target spacecraft using simulations. The four components of the proposed architecture, shown in Fig. 1, are: 1) information estimation, 2) state and inertial frame estimation, 3) guidance (or) motion planning, and 4) control. In this work, we assume that the state and inertial frame estimation algorithms are available and focus on motion planning for information maximization. The information estimation quantifies the information gain of the target spacecraft, given past and potential future poses of all the observer spacecraft. The estimated information gain is a crucial input to the motion planner, which considers a set of alternative PROs and decides when and how to reconfigure the observer spacecraft, while approximately maximizing the information gain and minimizing delta-V. For computational efficiency, both information estimation and motion planning modules rely on simplified attitude dynamics. Our controls component ensures PRO initialization, safe reconfiguration, and instrument pointing.

### A. Related Work

The prior research on autonomous on-orbit servicing can be classified into three lines of work: 1) vision-based navigation and mapping with control [14, 15], 2) formation maintenance control [16] for inspection, and 3) single spacecraft flight mission [4]. The main focus of the vision-based navigation [17] research has been to develop inspection and mapping algorithms using sensors such as monocular cameras and laser range finders. Using the navigation estimates, a pointing control and formation maintenance control was designed to ensure that the target body is within the field of view of the sensors. This work is applicable to single spacecraft missions and does not take into account the



**Fig. 1 Proposed multi-spacecraft GNC system architecture for autonomous inspection and mapping.**

overall formation maintenance and safety during the inspection task. We propose an architecture that can be integrated with any of the aforementioned navigation algorithms and extends to multi-spacecraft collaborative inspection by using an information metric to design stable relative orbits and minimum energy attitude motion plan for inspection.

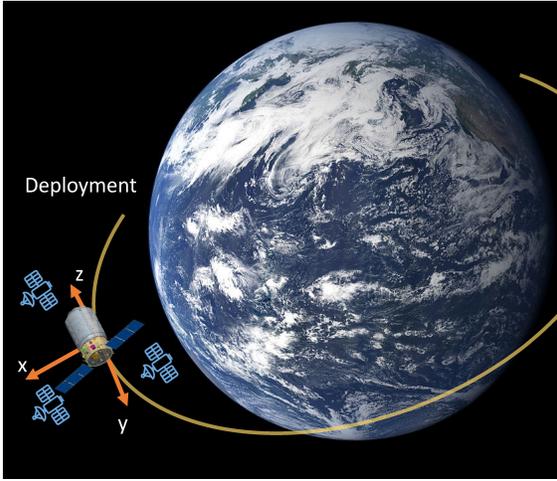
A very recent work considers a centralized approach [16] to integrate formation relative orbit planning and control for the inspection task but is limited to the scenario where the target spacecraft are continuously tracked by the observer spacecraft using attitude control. The algorithm presented in this paper is applicable to a wide range of scenarios when the full pose information about the observer spacecraft is available. Along with the work on algorithm development, the recent mission Seeker [4] demonstrated on-orbit inspection using a single observer to track and estimate the bounding box of the Cygnus spacecraft in both dark and light background with neural networks.

## B. Paper Organization

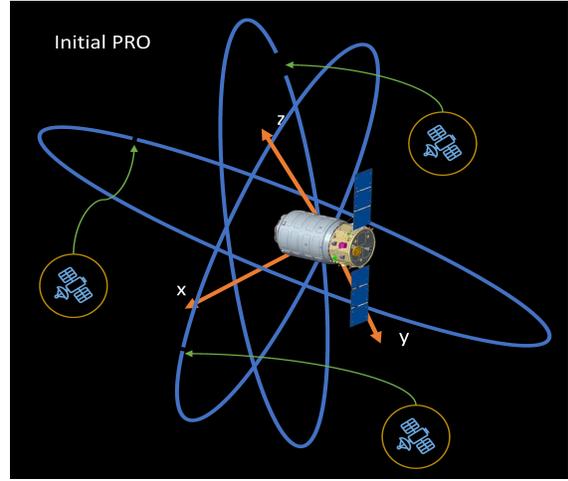
The paper is organized as follows. We discuss the problem formulation along with preliminaries on the relative orbital dynamics in Section II. The main algorithm along with extension to distributed inspection is presented in Section III. In Section IV, we present the distributed inspection algorithm for inspecting the target spacecraft model by computing and tracking optimal observation trajectories of multiple observer spacecraft. We conclude the paper in Section V with a brief discussion on the results of the proposed architecture.

## II. Problem Formulation

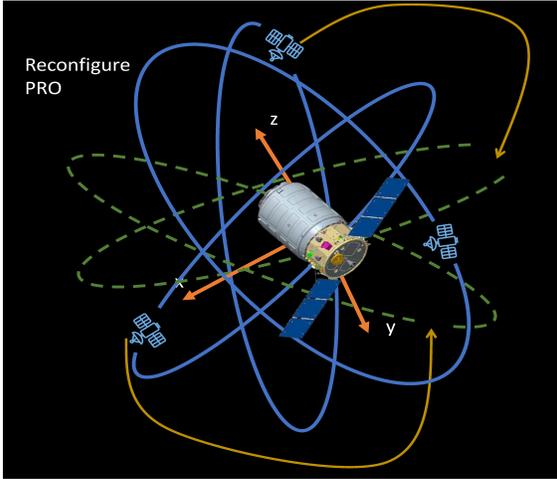
In this section, we discuss the concept of operations of a typical Earth orbit inspection mission with the target spacecraft as an example along with the preliminaries for the proposed architecture. The scenario considered in this paper has three phases, see Fig. 2. In the first phase, the small observer spacecraft are deployed from the target spacecraft and begin a drift phase. The drifting spacecraft are then inserted into a parking PRO or an initial PRO in the second phase. In the third phase, the spacecraft in stable relative orbits are used for inspecting the target. As needed, the spacecraft reconfigure to a new set of PROs to inspect a previously unobserved surface area on the target spacecraft. In this paper, we use the Hills-Clohessy-Wiltshire (HCW) equations to describe the relative orbital dynamics of the observer CubeSats. For the stable relative orbit initialization and reconfiguration phase, we formulate an optimal control problem with  $\mathcal{L}_1$  fuel cost and safety, energy matching as constraints, and solve it using sequential convex programming (SCP), similar to prior work [18]. The planned trajectories are tracked using a model predictive control formulation of the convexified problem. During the inspection phase, we represent the attitude dynamics using quaternions. The attitude planning is done using a combination of slerp interpolation and SCP with norm constraint on the quaternions. We use an existing nonlinear feedback controller for attitude tracking [19]. In the following, we briefly review the HCW equations, the energy matching condition for stable relative orbits, and convexification of optimal control problems for relative orbit and attitude motion planning.



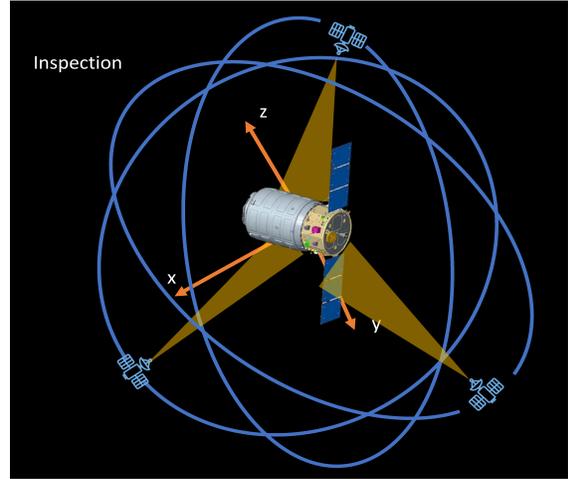
(a) Three observer spacecraft deployed from the target spacecraft.



(b) Initialization of the deployed observer spacecraft into stable relative orbits.



(c) Reconfiguration of the observer spacecraft to inspect a different surface area on the target.



(d) Pointing control for inspection of the target spacecraft.

**Fig. 2 Concept of operations of an inspection mission for observing a target spacecraft with multiple observer spacecraft deployed.**

### A. Relative Dynamics

The relative orbit dynamics [20, 21] of the observer spacecraft with respect to the target spacecraft in chief orbit are described in a Local-Vertical Local-Horizontal (LVLH) frame  $(x, y, z)$  as shown in Fig. 2a that is fixed to the target spacecraft. The  $x$  coordinate is in the chief orbit's radial direction, the  $z$  coordinate is defined in the angular momentum direction, and the  $y$  coordinate completes the right handed coordinate system. We assume that the target spacecraft's dynamics is defined by the Gauss's variational equation [22],

$$\dot{\mathbf{a}} = \mathbf{f}_c(\mathbf{a}) + \mathbf{G}(\mathbf{a})\mathbf{u}_c, \quad (1)$$

where  $\mathbf{a} \in \mathbb{R}^6$  is the orbital element vector in the ECI frame, the bias term  $\mathbf{f}_c$  determines the rate of change of  $\mathbf{a}$ , and  $\mathbf{G}$  is the control allocation matrix for the control input  $\mathbf{u}_c \in \mathbb{R}^3$ . In the LVLH frame, the functional form of the relative orbital dynamics of the  $i^{\text{th}}$  observer is defined as follows:

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i, \mathbf{a}, \dot{\mathbf{a}}, \mathbf{u}_i), \quad (2)$$

where the bias term  $\mathbf{f}_i$  depends on the target's dynamics  $\mathbf{a}$  and the force input  $\mathbf{u}_i$ . For the definitions of  $\mathbf{f}_c$ ,  $\mathbf{f}_i$ , and  $\mathbf{G}$  see [22]. We assume that the target spacecraft is in a circular orbit, the linearized relative dynamics  $\mathbf{f}_i$  of the  $i^{\text{th}}$  observer spacecraft for continuous control input  $u_i$  in the LVLH frame are as follows:

$$\ddot{x} - 2n\dot{y} - 3n^2x = \frac{f_{\text{thr}}}{m}u_x, \quad \ddot{y} + 2n\dot{x} = \frac{f_{\text{thr}}}{m}u_y, \quad \ddot{z} + n^2z = \frac{f_{\text{thr}}}{m}u_z, \quad (3)$$

where  $n = \sqrt{\mu/r_0^3}$  is the mean motion of the target spacecraft,  $\mu$  is the gravitational constant,  $r_0$  is the radius of the chief orbit with respect to the ECI frame, and  $m$  is the mass of the observer spacecraft. The control input  $\mathbf{u} = [u_x, u_y, u_z]^T$  scales the thruster force  $f_{\text{thr}}$  similar to a pulse-width modulated signal. These dynamics in Linear Time Invariant (LTI) state-space form are given by:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\frac{\mathbf{u}}{m}, \quad (4)$$

where  $\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$ . The matrices  $\mathbf{A}$  and  $\mathbf{B}$  are given in Eq. (40) (appendix).

The discrete-time dynamics for a time step  $T$  is given by:

$$\mathbf{x}[k+1] = \mathbf{A}_d\mathbf{x}[k] + \mathbf{B}_d\mathbf{u}[k], \quad \text{with} \quad \mathbf{A}_d = e^{\mathbf{A}\Delta t}, \quad \mathbf{B}_d = \int_{t=0}^T e^{\mathbf{A}\Delta t} \frac{\mathbf{B}}{m} dt, \quad (5)$$

where  $k$  is the current time-step, and  $\Delta t$  is the discretization time step. The matrix  $\mathbf{A}_d$  is given in Eq. (41). For continuous control  $\mathbf{u}(t)$  the input matrix  $\mathbf{B}_d$  is given in Eq. (42).

The LTI dynamics with control input  $\mathbf{u}$  is used in designing trajectories for reaching a stable relative orbit and for reconfiguration maneuvers to a new set of stable relative orbits. The relative orbits for parking and inspection are designed using the energy matching condition discussed below in Eq. (6), which results in passively stable orbits that require minimum fuel for maintenance.

The global criterion for stable relative orbits is given by the energy matching

$$\frac{1}{2} \left( (\dot{x} - ny)^2 + (\dot{y} + n(x + r_0))^2 + \dot{z}^2 \right) - \frac{\mu}{\sqrt{(r_0 + x)^2 + y^2 + z^2}} = -\frac{\mu}{2r_0}. \quad (6)$$

In Eq. (6), the energy of the observer is matched with that of the target spacecraft at all time  $t$ . Using this criteria to design the orbits of the deputies results in a bounded relative motion with respect to the LVLH frame. For simplified analysis, we use a subset of the energy matched orbits defined by the following initial condition constraint for the HCW dynamics when  $\mathbf{u} = 0$ :

$$\dot{y}_0 = -2nx_0, \quad (7)$$

where  $\dot{y}_0$  is the initial velocity in  $y$  direction and  $x_0$  is the initial position in the LVLH frame. The initial  $\Delta V$  in  $y_0$  is computed using the initial position information. We use this condition to generate a pool of stable relative trajectories and search for an optimal trajectory in the pool for conducting the inspection mission.

## B. Relative Orbit Initialization and Reconfiguration

The trajectory design for stable relative orbit initialization and orbit reconfiguration is formulated as a fixed-time optimal control problem with  $\mathcal{L}_1$  fuel cost. For  $N$  observer spacecraft the optimal control problem is given in the following problem.

### Problem 1 Centralized Continuous Time Relative Orbit Reconfiguration

$$\mathcal{J}_{\text{PRO}} = \min_{\mathbf{x}_i, \mathbf{u}_i, i \in \{1, \dots, N\}} \sum_{i=1}^N \int_{t_0}^{t_f} \|\mathbf{u}_i(t)\|_1 dt \quad (8)$$

$$\text{s.t. } \dot{\mathbf{x}}_i = \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}_i \quad \forall t \in [t_0, t_f] \quad (9)$$

$$\|\mathbf{C}(\mathbf{x}_i - \mathbf{x}_j)\|_2 \geq r_c \quad \forall t \in [t_0, t_f], \quad j > i, \quad j = \{2, \dots, N\} \quad (10)$$

$$\mathbf{u}_i \in \mathcal{U} \quad \forall t \in [t_0, t_f] \quad (11)$$

$$\mathbf{x}_i(t_0) = \mathbf{x}_{i0} \quad \& \quad \mathbf{x}_i(t_f) = \mathbf{x}_{if} \quad (12)$$

where  $i$  denotes the observer and  $i \in \{1, \dots, N\}$ ,  $\mathbf{C} = [\mathbf{I}_{3 \times 3} \ 0_{3 \times 3}]$ ,  $r_c$  is the minimum safe allowable distance between two spacecraft,  $\mathcal{U}$  is a convex control constraint set, and  $\mathbf{x}_{i0}$  and  $\mathbf{x}_{if}$  are the initial and terminal conditions of the  $i^{\text{th}}$  observer. The optimal control problem is transformed to a convex optimization problem by using the discrete dynamics in Eq. (5) and convexifying the collision constraint in Eq. (10) about a nominal trajectory  $(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i)$ , given the full trajectory  $(\bar{\mathbf{x}}_j, \bar{\mathbf{u}}_j)$  of the neighboring observer spacecraft  $j$ , as discussed in prior work [23, 24]. For each observer spacecraft  $i$  the following decentralized convex optimization problem is solved to compute the trajectory.

**Problem 2** *Decentralized Discrete Time Convex Relative Orbit Reconfiguration*

$$\mathcal{J}_{\text{PRO}_i} = \min_{\mathbf{x}_i, \mathbf{u}_i} \sum_{k=0}^{T-1} \|\mathbf{u}_i[k]\|_1 \Delta t \quad (13)$$

$$\text{s.t. } \mathbf{x}_i[k+1] = \mathbf{A}_d \mathbf{x}_i[k] + \mathbf{B}_d \mathbf{u}_i[k] \quad \forall k \in \{1, \dots, T-1\} \quad (14)$$

$$\begin{aligned} & (\bar{\mathbf{x}}_i[k] - \bar{\mathbf{x}}_j[k])^\top \mathbf{C}^\top \mathbf{C} (\mathbf{x}_i[k] - \bar{\mathbf{x}}_j[k]) \geq r_c \|\mathbf{C}(\bar{\mathbf{x}}_i[k] - \bar{\mathbf{x}}_j[k])\|_2 \\ & \forall k \in \{0, 1, \dots, T\}, \quad j > i, \quad j = \{2, \dots, N\} \end{aligned} \quad (15)$$

$$\mathbf{u}_i[k] \in \mathcal{U} \quad \forall k \in \{0, 1, \dots, T-1\} \quad (16)$$

$$\mathbf{x}_i[0] = \mathbf{x}_{i0} \quad \& \quad \mathbf{x}_i[T] = \mathbf{x}_{if} \quad (17)$$

where  $T$  is the number of time steps in the time interval  $[t_0, t_f]$ ,  $k=0$  and  $k=T$  correspond to the initial condition and terminal condition, respectively. Problem 2 is used to compute a safe trajectory for stable relative orbit initialization and reconfiguration. The choice of initial and terminal conditions for each phase of the mission in Fig. 2 is discussed in Section III. The convex program is initialized with a straight line trajectory from initial to terminal conditions as the nominal trajectory. A *model predictive control* (MPC) formulation of the above problem can be used for controlling the spacecraft to track the designed trajectory [23, 24]. The orbital dynamics and attitude dynamics are assumed to be decoupled for the motion planning and control design. We discuss the attitude dynamics and trajectory design problem formulation in the following sections.

### C. Attitude Dynamics

The attitude of the spacecraft can be represented by any attitude representations including quaternions [25], Modified Rodrigues Parameters (MRPs) [25], and SO(3) rotation matrices. For the attitude planning problem we use quaternions  $\mathbf{q} \in \mathbb{R}^4$  and  $\|\mathbf{q}\|_2 = 1$ ,  $\mathbf{q} \in \mathbb{H}$ , where  $\mathbb{H}$  is the Hamiltonian space and have a bijective mapping to the quaternion sphere. The attitude kinematics equation is given using the body angular rates  $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^\top \in \mathbb{R}^3$ . The kinematics of quaternions are given as

$$\dot{\mathbf{q}} = \boldsymbol{\Omega}(\boldsymbol{\omega})\mathbf{q}, \quad (18)$$

where  $\boldsymbol{\Omega}(\boldsymbol{\omega})$  is given in Eq. (43). The attitude dynamics is as follows:

$$\dot{\boldsymbol{\omega}} = -\mathbf{J}^{-1}(\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}) + \mathbf{J}^{-1}\boldsymbol{\tau}, \quad (19)$$

where  $\mathbf{J}$  is inertia tensor of the observer,  $\boldsymbol{\tau}$  is the input torque, and  $\times$  denotes the vector cross product. The continuous dynamics is linearized around a nominal trajectory  $(\bar{\mathbf{q}}, \bar{\boldsymbol{\omega}})$  and the functional form of the linear system is defined in the following equations:

$$\dot{\mathbf{q}} = \mathbf{A}_{1q}(\mathbf{q} - \bar{\mathbf{q}}) + \mathbf{A}_{1\omega}(\boldsymbol{\omega} - \bar{\boldsymbol{\omega}}) + \boldsymbol{\Omega}(\bar{\boldsymbol{\omega}})\bar{\mathbf{q}}, \quad \text{where } \mathbf{A}_{1q} = \left. \frac{\partial \boldsymbol{\Omega} \mathbf{q}}{\partial \mathbf{q}} \right|_{(\mathbf{q}, \boldsymbol{\omega}) = (\bar{\mathbf{q}}, \bar{\boldsymbol{\omega}})}, \mathbf{A}_{1\omega} = \left. \frac{\partial \boldsymbol{\Omega} \mathbf{q}}{\partial \boldsymbol{\omega}} \right|_{(\mathbf{q}, \boldsymbol{\omega}) = (\bar{\mathbf{q}}, \bar{\boldsymbol{\omega}})} \quad (20)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{A}_{2\omega}(\boldsymbol{\omega} - \bar{\boldsymbol{\omega}}) + \mathbf{J}^{-1}\boldsymbol{\tau} - \mathbf{J}^{-1}(\bar{\boldsymbol{\omega}} \times \mathbf{J}\bar{\boldsymbol{\omega}}), \quad \text{where } \mathbf{A}_{2\omega} = - \left. \frac{\partial (\mathbf{J}^{-1}(\bar{\boldsymbol{\omega}} \times \mathbf{J}\bar{\boldsymbol{\omega}}))}{\partial \boldsymbol{\omega}} \right|_{\boldsymbol{\omega} = \bar{\boldsymbol{\omega}}}. \quad (21)$$

The linearized equations are discretized for a time interval  $\Delta t_a$  and time steps  $T_a$  and represented as follows to use in the sequential convex programming formulation:

$$\mathbf{q}[k+1] = \mathbf{A}_{1q}(\bar{\mathbf{q}}, \bar{\boldsymbol{\omega}}, \Delta t_a)[k]\mathbf{q}[k] + \mathbf{A}_{1\omega}(\bar{\mathbf{q}}, \bar{\boldsymbol{\omega}}, \Delta t_a)[k]\boldsymbol{\omega}[k] + \mathbf{C}_1(\bar{\mathbf{q}}, \bar{\boldsymbol{\omega}}, \Delta t_a)[k], \quad (22)$$

$$\boldsymbol{\omega}[k+1] = \mathbf{A}_{2\omega}(\bar{\boldsymbol{\omega}}, \Delta t_a)[k]\boldsymbol{\omega}[k] + \mathbf{J}^{-1}\boldsymbol{\tau}[k] + \mathbf{C}_2(\bar{\boldsymbol{\omega}}, \Delta t_a)[k], \quad (23)$$

where  $k \in \{1, \dots, T_a\}$  is the time step. The matrices  $\mathbf{A}_{1q}$ ,  $\mathbf{A}_{1\omega}$ ,  $\mathbf{C}_1$ ,  $\mathbf{A}_{2\omega}$ , and  $\mathbf{C}_2$  are computed using the symbolic Python package SymPy. The linear, discrete form of the attitude dynamics is used to construct a sequential convex programming problem for designing a smooth and optimal attitude trajectory.

#### D. Attitude Planning Problem

The attitude planning problem is formulated as the following optimal control problem with a norm constraint on the quaternions:

##### Problem 3 Continuous Time Attitude Planning

$$\begin{aligned}
\mathcal{J}_{\text{attitude}} &= \min_{\mathbf{q}, \boldsymbol{\omega}, \boldsymbol{\tau}} \int_{t_0}^{t_f} \|\boldsymbol{\tau}(t)\|_2 dt \\
\text{s.t. } \dot{\mathbf{q}} &= \boldsymbol{\Omega}(\boldsymbol{\omega})\mathbf{q} \quad \forall t \in [t_0, t_f] \\
\dot{\boldsymbol{\omega}} &= -\mathbf{J}^{-1}(\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}) + \mathbf{J}^{-1}\boldsymbol{\tau} \quad \forall t \in [t_0, t_f] \\
\mathbf{q}^\top \mathbf{q} &= 1 \quad \forall t \in [t_0, t_f] \\
\|\boldsymbol{\tau}\|_\infty &\leq \tau_{\max} \quad \forall t \in [t_0, t_f] \\
\mathbf{q}(t_0) &= \mathbf{q}_0, \boldsymbol{\omega}(t_0) = \boldsymbol{\omega}_0 \quad \& \quad \mathbf{q}(t_f) = \mathbf{q}_f, \boldsymbol{\omega}(t_f) = \boldsymbol{\omega}_f,
\end{aligned} \tag{24}$$

where  $\tau_{\max}$  is the maximum torque that can be applied on the spacecraft using reaction wheels, and  $(\mathbf{q}_0, \boldsymbol{\omega}_0)$  and  $(\mathbf{q}_f, \boldsymbol{\omega}_f)$  are the initial and terminal orientation and angular velocity, respectively. The solution of the optimal control Problem 3 is a minimum energy attitude maneuver from initial orientation  $\mathbf{q}_0$  to a terminal orientation  $\mathbf{q}_f$ . We use the discrete dynamics in Eqs. (22) and (23) to formulate the following convex optimization problem given a nominal attitude  $\bar{\mathbf{q}}$  and angular velocity trajectory  $\bar{\boldsymbol{\omega}}$ .

##### Problem 4 Linearized Discrete Time Attitude Planning

$$\begin{aligned}
\mathcal{J}_{\text{attitude}} &= \min_{\mathbf{q}, \boldsymbol{\omega}, \boldsymbol{\tau}} \sum_{k=0}^{T_a} \|\boldsymbol{\tau}[k]\|_2 \Delta t_a \\
\text{s.t. } & \text{Eqs. (22) and (23)} \quad \forall k \in \{0, \dots, T_a - 1\} \\
\bar{\mathbf{q}}[k]^\top \mathbf{q}[k] &= 1 \quad k \in \{0, \dots, T_a - 1\} \\
\boldsymbol{\tau}[k+1] - \boldsymbol{\tau}[k] &\leq \alpha \Delta t_a \quad k \in \{0, \dots, T_a - 2\} \\
\|\boldsymbol{\tau}[k]\|_\infty &\leq \tau_{\max} \quad k \in \{0, \dots, T_a\} \\
\mathbf{q}[0] &= \mathbf{q}_0, \boldsymbol{\omega}[0] = \boldsymbol{\omega}_0 \quad \& \quad \mathbf{q}[T_a] = \mathbf{q}_f, \boldsymbol{\omega}[T_a] = \boldsymbol{\omega}_f
\end{aligned} \tag{25}$$

The norm constraint  $\|\mathbf{q}\|_2 = 1$  is convexified by quasi-linearization around the nominal trajectory  $\bar{\mathbf{q}}$ , leading to the linear equality constraint  $\bar{\mathbf{q}}^\top \mathbf{q} = 1$ . We additionally introduce the constraint in Eq. (28) to account for the actuator dynamics, where  $\alpha$  is a constant depending on the input signal to the actuators. For a given initial and terminal quaternion  $\mathbf{q}_0$  and  $\mathbf{q}_f$ , we initialize Problem 4 by computing the nominal quaternion  $\bar{\mathbf{q}}$  trajectory using slerp interpolation. The nominal angular velocity trajectory  $\bar{\boldsymbol{\omega}}$  is computed using  $\bar{\mathbf{q}}$  and the kinematics in Eq. (18). Problem 4 is solved using sequential convex programming. The attitude motion plan is tracked using the nonlinear control law derived in prior work [19]. In this paper, we integrate Problem 2, Problem 4, and control algorithms [19, 23] with an information cost and derive an algorithm for active inspection of a spacecraft.

### III. Main Distributed Inspection Algorithm

Our approach has four major components: information estimation, state estimation, motion planning, and control, see Fig. 1. The information estimation quantifies the information gain of the target spacecraft, given past and potential future poses of all the observer spacecraft. The estimated information gain is a crucial input to the motion planner, which considers a set of alternative *passive relative orbits* (PROs) and decides when and how to reconfigure the spacecraft, while approximately maximizing the information gain and minimizing the control effort. For computational efficiency, both information estimation and motion planning modules rely on simplified attitude dynamics. Our controls component tracks the planned motions at runtime. We assume that an accurate estimate of the LVLH frame and the full state of the observer spacecraft in the LVLH frame and the target spacecraft in the ECI frame are available at all times during the guidance and control.

We first discuss the centralized implementation of the algorithm and then discuss on the extension to distributed architecture with communication between observers. The algorithm is implemented using a hierarchical framework, where different components are executed at different frequencies, see Fig. 3. In an offline pre-processing stage, we use a

rough prior model of the target spacecraft to generate PRO candidates as well as a ray casting database, which is a data structure that enables efficient computation of the information gain at runtime. Our motion planning module runs at a low rate  $\omega_{low}$ , e.g., once per orbit. It uses the PRO candidates, ray casting database, as well as information database to decide whether and how the spacecraft should reconfigure to different PROs. These PROs are maintained using our PRO controller at a mid-rate  $\omega_{mid}$ , e.g., every few seconds. At the same timescale, we plan for new desired attitudes, take pictures of the target spacecraft, and update our relative state estimate and information cost. At the highest rate  $\omega_{high}$ , e.g., several times per second, we use attitude trajectory optimization and control. The pseudo code of our approach is given in Algorithm 1 for centralized implementation.

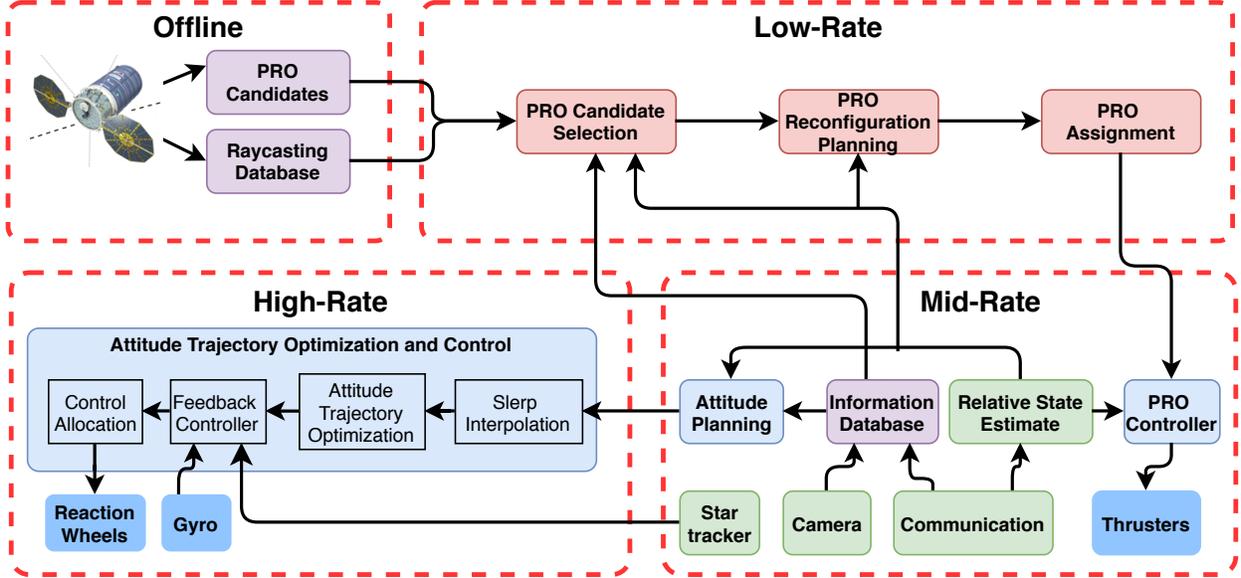


Fig. 3 The multi-level and multi-timescale hierarchical architecture for combined orbit and attitude planning and control for an inspection task.

## A. Prerequisites: Information Metric

### 1. Information Gain

To quantify the information, we assume that we have a (rough) prior model of the target spacecraft and sample *points of interest* (POIs) on its surface. The overall goal is then to minimize the variance of the estimation of all POIs. Specifically, we introduce the *information cost*,  $H$ , based on prior work [26], as follows:

$$H_{\text{POI}}(\mathbf{s}) = \left( w^{-1} + \sum_{\mathbf{p} \in \mathcal{P}} f(\mathbf{p}, \mathbf{s})^{-1} \right)^{-1} \quad (31)$$

$$H = \sum_{\mathbf{s} \in \text{POIs}} H_{\text{POI}}(\mathbf{s}) \phi(\mathbf{s}), \quad (32)$$

where  $\mathbf{s} \in \mathbb{R}^3$  is a sampled POI on the target spacecraft's surface,  $w \in \mathbb{R}$  is the basic variance based on the prior model of the target spacecraft,  $\mathbf{p} \in SE(3)$  is the pose of a sensor mounted on a spacecraft such as a camera,  $\mathcal{P}$  is the set of all sensor poses,  $f(\mathbf{p}, \mathbf{s})$  estimates the variance of estimating POI at  $\mathbf{s}$  with the sensor at  $\mathbf{p}$ , and  $\phi(\mathbf{s}) \in \mathbb{R}$  is the relative importance of POI  $\mathbf{s}$ . The cost  $H$  is similar to Bayesian sensor fusion, where the estimates are fused using the confidence in the noisy sensor outputs. This formulation can handle important cases such as multiple spacecraft observing the same POI well. Additionally, minimizing  $H$  also minimizes the expected variance of the estimation of the POIs [26].

For a camera, the function  $f(\cdot, \cdot)$  corresponds to information per pixel. Minimizing  $f$  entails computing the location and orientation of the camera to optimally capture a POI. The choice of  $f$  can directly incorporate other sensor or spacecraft characteristics, such as the current uncertainty of the spacecraft's pose estimate, the accuracy of the sensor

---

**Algorithm 1** Multi-level Hierarchical Guidance and Control Architecture

---

- 1: **Input 1:** Prior rough model of the target, points of interest (POI) on its surface, and the number of observers  $N$
  - 2: **Input 2:** Initialized instance of the *information cost* (Section III.A.1) and *information database* (Section III.A.2)
  - 3: **Input 3:** Minimum safe distance  $d_{\min}$  and desired sensing distance  $d_s$  from the observer to the target, sensor field of view
  - 4: **Input 4:** Pose estimates of all the observers and the LVLH frame available to all the observers at all time
- 

**Pre-processing [Offline]**

---

- 5: **procedure** PREPROCESSING(Inputs 1, 2, 3, 4)
  - 6:     Generate *PRO candidate* pool as discussed in Section III.B.1
  - 7:     Generate *ray casting database*
  - 8:     Initialize *information database* using prior model
  - 9:     **return** *PRO candidates, information database, ray casting database*
  - 10: **end procedure**
- 

**Low-Rate [loop frequency:  $\omega_{low}$ ]**

---

- 11: **procedure** LOWRATE(*PRO candidates, information database, ray casting database*)
  - 12:     Generate an optimal set of  $N$  PROs that minimizes *information cost*, see Section III.C.1
  - 13:     **for each** *observer* in  $\{1, \dots, N\}$  **do**
  - 14:         Compute  $N$  *reconfiguration plans* for the *observer* to each selected PRO, see Section III.C.2
  - 15:     **end for**
  - 16:     Compute optimal *PRO assignment* to each observer that minimizes total reconfiguration cost, see Section III.C.3
  - 17:     **return** *reconfiguration plan, PRO assignment*
  - 18: **end procedure**
- 

**Mid-Rate [loop frequency:  $\omega_{mid}$ ]**

---

- 19: **procedure** MIDRATE(*reconfiguration plan, PRO assignment, information database, ray casting database*)  $\triangleright$  The PRO initialization and reconfiguration can be done using decentralized control.
  - 20:     **for each** *observer* in  $\{1, \dots, N\}$  **do**
  - 21:         **if** *reconfiguration plan* is available **then**
  - 22:             Execute *reconfiguration plan* using MPC controller discussed in [23]
  - 23:             Plan attitude using visibility check with the *ray casting database*
  - 24:             *information update*, see Section III.D.1  $\triangleright$  Communicate *information update* to the observers.
  - 25:         **else**
  - 26:             Propagate the assigned PRO, HCW dynamics using Eq. (7)
  - 27:             Maintain Stable HCW using *PRO controller*, see Section III.D.3
  - 28:             Plan attitude using visibility check with the *ray casting database*
  - 29:             *information update*, see Section III.D.1  $\triangleright$  Communicate *information update* to the observers.
  - 30:         **end if**
  - 31:     **end for**
  - 32:     **return** *attitude plan, information update*
  - 33: **end procedure**
- 

**High-Rate [loop frequency:  $\omega_{high}$ ]**

---

- 34: **procedure** HIGHRATE(*attitude plan*)
  - 35:     **for each** *observer* in  $\{1, \dots, N\}$  **do**
  - 36:         Attitude trajectory optimization using Problem 4
  - 37:         Nonlinear attitude control using the smooth trajectory as reference input [19]
  - 38:     **end for**
  - 39: **end procedure**
-

based on the distance between  $\mathbf{p}$  and  $\mathbf{s}$ , or the lighting conditions. Here, we use a simple formulation of  $f$  assuming an RGB camera sensor and no environmental noise [27]:

$$f(\mathbf{p}, \mathbf{s}) \propto \begin{cases} \text{dist}^2(\mathbf{p}, \mathbf{s}) & \mathbf{s} \text{ visible from } \mathbf{p} \\ \infty & \text{otherwise} \end{cases}, \quad (33)$$

where  $\text{dist}(\mathbf{p}, \mathbf{s})$  is the Euclidean distance between POI  $\mathbf{s}$  and pose  $\mathbf{p}$ . In practice, computing  $f$  requires an efficient visibility check. Since we have a rough prior model of the target spacecraft, we propose to use a *ray casting database*, which uses pre-processing to speed up visibility checks significantly, see Section III.B.2.

## 2. Information Database

Let  $\mathcal{P}$  be the set of all sensor poses. For each POI  $\mathbf{s}$ , we store the set of sensor poses  $\mathbf{p} \in \mathcal{P}$  that can observe  $\mathbf{s}$  as  $\mathcal{P}_{\text{POI}}(\mathbf{s})$ , i.e.:

$$\mathcal{P}_{\text{POI}}(\mathbf{s}) = \{\mathbf{p} \in \mathcal{P} | f(\mathbf{p}, \mathbf{s}) \text{ finite}\}. \quad (34)$$

We call the data structure that stores  $\mathcal{P}_{\text{POI}}(\mathbf{s})$  for all  $\mathbf{s} \in \text{POIs}$  *information database*. We use the fact that our state estimator uses the common LVLH frame for all the observers, making direct pose exchanges feasible. The information database is compact and can be stored locally on each observer. Furthermore, if the covariance of the pose estimate is available, it can be communicated and used as well for an improved uncertainty sensor model  $f$ .

## B. Pre-processing: PRO Candidates and Ray Casting

### 1. PRO Candidate Computation

Passive Relative Orbits (PROs) play an important role in the proposed GNC architecture as it provides thrust-free orbits around the target spacecraft that can be utilized for on-orbit inspection and mapping. With prior knowledge of the target spacecraft model and the desired sensing distance of a given sensor, we can pre-compute a finite set of PRO candidates that will be utilized throughout the on-orbit inspection and mapping in real-time.

To generate a set of PRO candidates to geometrically cover the target spacecraft from varying perspectives in a systematic manner, we first express the solution to HCW Eq. (3) from Cartesian to Phase-Magnitude [20] form:

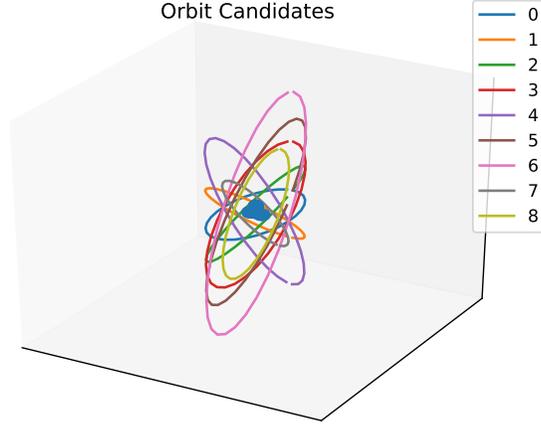
$$x(t) = \rho_x \sin(nt + \alpha_x), \quad y(t) = \rho_y + 2\rho_x \cos(nt + \alpha_x), \quad z(t) = \rho_z \sin(nt + \alpha_z) \quad (35)$$

$$\rho_x = \frac{\sqrt{\dot{x}_0^2 + x_0^2 n^2}}{n}, \quad \rho_y = y_0 - 2\dot{x}_0/n, \quad \rho_z = \frac{\sqrt{\dot{z}_0^2 + z_0^2 n^2}}{n} \quad (36)$$

$$\alpha_x = \tan^{-1} \frac{n x_0}{\dot{x}_0}, \quad \alpha_z = \tan^{-1} \frac{n z_0}{\dot{z}_0}. \quad (37)$$

Then we set  $\rho_y = 0$  to restrict the PRO candidates to be concentric around the target spacecraft. The concentric PROs ensure collision avoidance [22] between spacecraft as multiple orbits around the target spacecraft are used for inspection. We further confine the PRO candidates to have orbits whose distance to the target spacecraft varies around the desired sensing distance of the sensor  $d_s$  by setting  $\rho_x = 0.5d_s$ . Note that these concentric PROs will have semi-major axis of  $d_s$  and semiminor axis of  $0.5d_s$  in its  $x - y$  projection. Then the parameter  $\rho_z$  and  $\alpha_z$  are varied from 0 to  $2d_s$  and  $-\pi/2$  to  $\pi/2$ , respectively, to generate PRO candidates to be utilized for the on-orbit inspection. The parameter  $\rho_z$  controls the eccentricity of the orbit and determines how far stretched a PRO is along the  $z$ -axis. The parameter  $\alpha_z$  rotates a PRO about the  $z$ -axis and provides varying cross-sectional view of the target spacecraft. The PROs generated in this manner are essentially 3D ellipses that are different cross-sections of a 3D cylinder encompassing the target spacecraft. PROs generated using this technique cover only the radial surface of a cylinder around the target spacecraft. In order to generate the PROs that cover top and bottom surfaces of the target spacecraft, we set the parameter  $\rho_z = d_s$ ,  $\rho_y = 0$ ,  $\alpha_x = 0$ , and vary  $\rho_x$  and  $\alpha_z$  in the range  $[d_{\min}, 0.5d_s]$  and  $[-\pi/2, \pi/2]$ , respectively. The parameter  $d_{\min}$  is the minimum safe distance to the target spacecraft. An example of generated PROs is shown in Fig. 4. Further, for each sampled PRO on the 3D cylinder, we compute the delta-V required to initialize stable PRO at that location.

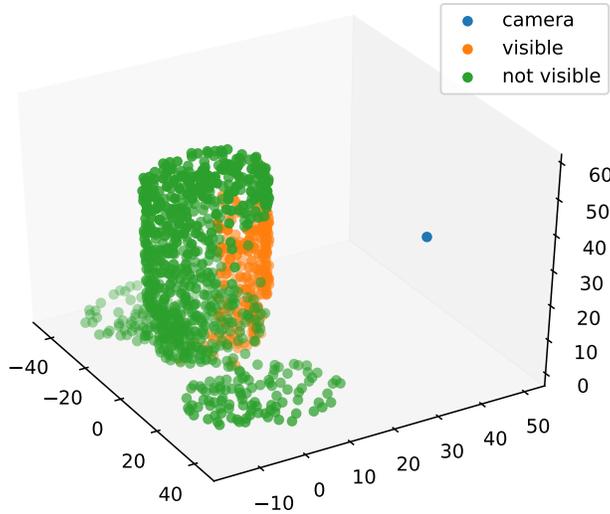
The total number of PRO candidates to be pre-computed will depend on the computational resources available on the spacecraft. The cardinality of the PRO candidate set should be large enough to be able to densely cover the target spacecraft, but not too large to enable online computation of PRO reconfiguration and assignment planning with on-board computation in real-time.



**Fig. 4** Nine generated PRO candidates to cover the target spacecraft (blue in the center).

## 2. Ray Casting Database

In practice, computing  $f$  requires an efficient visibility check. Since we have a rough prior model of the target spacecraft, we propose to use a *ray casting database*, which uses pre-processing to speed up visibility checks significantly. First, we use the model to sample candidate points of interest (POIs). Then, we i) uniformly create potential rays, ii) for each POI we use ray casting to find the subset of the potential rays that are not occluded, and iii) we store the result in a hash map indexed by spherical coordinates of the rays for fast look-up. At runtime, we first convert the position of  $p$  to spherical coordinates and use bilinear mapping of the nearby candidate rays to determine visibility. Second, we consider the attitude of  $p$  and the field of view of the sensor. An example of the visibility check for a fixed camera is shown in Fig. 5.



**Fig. 5** An example of the visibility check using ray casting database for a given camera with fixed location, pointed in  $-y$  direction with a field of view of 30 degrees.

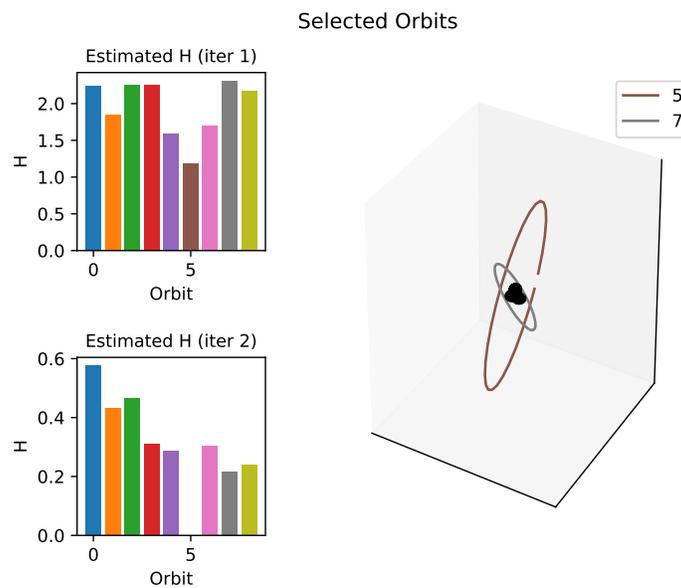
## C. Low-Rate: Orbit Motion Planning

The orbit motion planner uses the *PRO candidates*, a set of pre-computed passive relative orbits, and decides when and how each each of the observer spacecraft should switch to one of the orbits in that candidate set. Our approach uses three parts: i) *PRO candidate selection* based on minimizing the information cost  $H$ , ii) *PRO reconfiguration planning* to minimize delta-V, and iii) optimal *PRO assignment*.

### 1. PRO Candidate Selection

For the *PRO candidate selection*, we use a sequential greedy approach to generate a PRO set from the PRO candidate pool that will collectively minimize the *information cost*  $H$  when its orbits are committed by the observers. For each PRO candidate, we first compute an estimate of  $H$  that would be attained if an observer spacecraft takes measurements along the PRO. The estimation of  $H$  is computed by considering future sensor poses sampled from the candidate PRO with attitudes greedily assigned, together with the past sensor poses from all observers in the *information database*. Then, the candidate PRO with the lowest estimate of  $H$  is appended as an element of the PRO set. The procedure is repeated  $N$  times such that the cardinality of the PRO set generated is equal to the number of observers.

In our proposed multi-spacecraft GNC architecture, the sequential greedy approach is favored over combinatorial optimization for its computational efficiency. Selecting a PRO set that minimizes the *information cost*  $H$  with optimality guarantee requires combinatorial computation. This approach has factorial scaling as the number of observers increase. In contrast, the sequential greedy approach scales linearly with  $N$  and is able to generate near-optimal solution whose optimality gap is sufficiently small in practice. Figure 6 demonstrates the *PRO candidate selection* using the sequential greedy approach and shows the resulting PRO set.



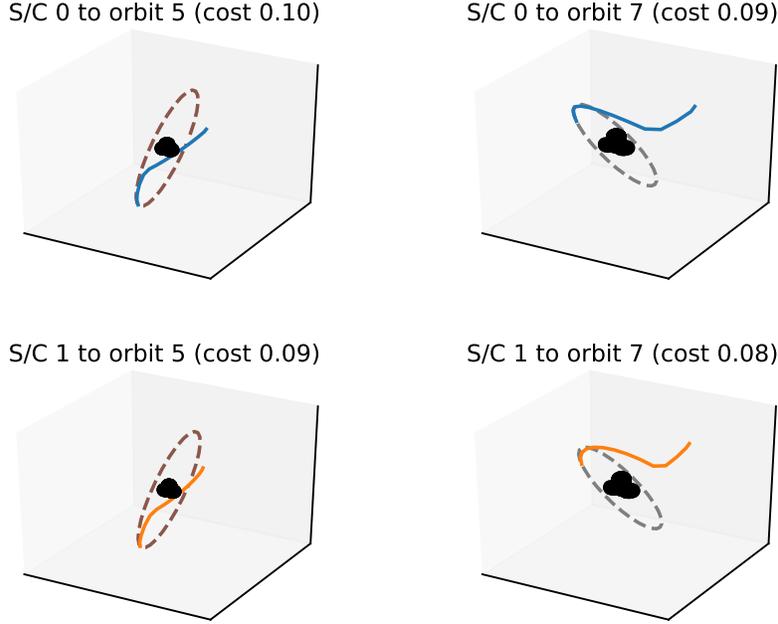
**Fig. 6** PRO candidate selection for  $N = 2$  from the set of PROs shown in Fig. 4. For each of the nine candidates the expected information cost is computed (bar chart top left). PRO 5 achieves the lowest  $H$  and is selected in the first iteration. Assuming one observer is orbiting PRO 5, we estimate  $H$  for the remaining PROs in the second iteration. The lowest cost is achieved for PRO 7 (bar chart bottom left). Observer spacecraft on the selected PROs can observe the target from many perspectives (plot on the right).

### 2. PRO Reconfiguration Planning

For *PRO reconfiguration planning* we use sequential convex programming in Problem 1 to compute minimum fuel trajectories that transfer an observer spacecraft from their current pose to each of the  $N$  selected PROs previously generated. In computing an optimal transfer trajectory for an observer spacecraft to a target PRO, the observer solves multiple instances of Problem 1 with varying terminal conditions Eq. (17) sampled along the target PRO. The lowest cost solution from the multiple problem instances forms the optimal transfer trajectory, and its corresponding estimate of the delta-V is recorded. This process is performed for each of the observers and can be easily done in a distributed manner.

A similar approach is utilized at the beginning of the mission for *PRO initialization*, which can be achieved by computing the initial thruster firing using the energy matching conditions derived in [22]. These initial conditions are shown to be robust to  $J_2$  perturbations.

## PRO Reconfiguration Planning



**Fig. 7** Example solutions for the PRO reconfiguration planning problem. Each of the two observers plan to move from its current state to one of the selected PRO candidates (see Fig. 6) and simultaneously estimates the delta-V cost for such maneuver (see captions on each plot).

### 3. PRO Assignment

For the *PRO assignment*, we use the Hungarian method [28] to assign candidate PROs to specific spacecraft, such that the estimated delta-V is minimized. For example, given 3 observers and 4 candidate PROs, the assignment is done such that  $\sum_{i=1}^3 \int_{t_0}^{t_f} \|u_i\|_1 dt$  is minimized. This PRO is initialized using the SCP formulation or by using the stable initial conditions through the reconfiguration planning. Such an approach can be easily distributed by utilizing, for example, distributed auctions [18].

In the example in Fig. 7 there are two possible assignments: observer 0 selects orbit 5 and observer 1 selects orbit 7, or observer 0 selects orbit 7 and observer 1 selects orbit 5. The Hungarian method computes the optimal assignment efficiently in polynomial time. Here, both costs are equal (0.18) and both assignments are optimal.

## D. Mid-Rate: Attitude Planning, Information Update, and PRO Controller

### 1. Information Update

We update our current information gain  $H$  as well as the *information database* in case of two events: if a new sensor measurement was obtained or if we receive a message from another observer of a new measurement. Let  $\mathbf{p}' \in SE(3)$  the new sensor pose. Then we can update the information database by i) finding the set of visible POIs from  $\mathbf{p}'$  using the ray casting database; and ii) updating the database:

$$\mathcal{P}_{\text{POI}}(\mathbf{s}) = \mathcal{P}_{\text{POI}}(\mathbf{s}) \cup \{\mathbf{p}'\}; \quad \forall \mathbf{s} \in \text{POIs s.t. } \mathbf{s} \text{ visible from } \mathbf{p}'. \quad (38)$$

Afterwards, we can re-compute  $H$  by evaluating (32). If a new measurement was obtained on the spacecraft,  $\mathbf{p}'$  is broadcasted to all other observers.

We note that  $H$  can also be re-evaluated incrementally more efficiently in the common case of  $\mathbf{p}'$  only observing a small subset of the POIs, because only a few  $H_{\text{POI}}(\mathbf{s})$  values will change. Furthermore, if  $f(\cdot, \cdot)$  depends on the covariance of  $\mathbf{p}'$ , the covariance needs to be included in the information database and communicated to the neighboring observers.

## 2. Attitude Planner

While the orbit motion planner (Section III.C) already considers attitudes to select the subset of suitable PROs, these attitudes might be impossible to track or are simply outdated based on the actual observations made so far. Therefore, we employ an efficient, greedy attitude planner algorithm, which takes the current value of  $H$  as well as angular velocity limits into account. A greedy approach is favored over more elaborate attitude planning, such as ones using dynamic programming [29], for computational efficiency.

Specifically, we try to point our camera towards an observable POI  $\mathbf{s}^*$  with the highest variance:

$$\mathbf{s}^* = \arg \max \{H_{\text{POI}}(\mathbf{s}) | \mathbf{s} \in \text{POIs and } \mathbf{s} \text{ visible from } \mathbf{p}\}, \quad (39)$$

where  $\mathbf{p}$  is the current pose of the observer. As before, the visibility check can be done efficiently using the ray casting database. Based on  $\mathbf{s}^*$ , the desired attitude can be computed such that  $\mathbf{s}^*$  is in the field of view.

## 3. PRO Controller

During long-term operation, the nonlinear effects like  $J_2$ , atmospheric drag, and the uncertainty in the thruster firing leads to drift in the spacecraft position with respect to the target spacecraft. To mitigate this we can perform *regular orbit maintenance* recomputing the energy matched delta-V using Eq. (7) and applying it to the spacecraft (or) by using a feedback tracking controller. We assume that the target spacecraft is on a circular orbit and use HCW equations that do not consider nonlinear ( $J_2$ ) and non-conservative effects (drag). This simplification allows us to develop an architecture that can be extend to include the nonlinear effects using initial conditions [22] that are robust to ( $J_2$ ) effect. For reconfiguration control, we use a model predictive control (MPC) [23] formulation of SCP by computing the control at fixed time intervals.

## E. High-Rate: Attitude Trajectory Optimization and Control

The attitude planner in Section III.D.2 generates a sequence of POIs for each spacecraft to point to in a greedy manner. The sequence of POIs are interpolated to a smooth attitude trajectory that is used as reference for the attitude tracking controller by solving Problem 4 using SCP. The optimization includes control constraints and actuator dynamics to ensure feasibility during tracking. The *attitude controller* tracks an interpolated attitude profile. The tracking of the interpolated attitude profile uses an exponentially stable controller [19]. The gains of the controller are tuned for the rise time and potential noise observed in the mission scenario.

## F. Properties and Remarks

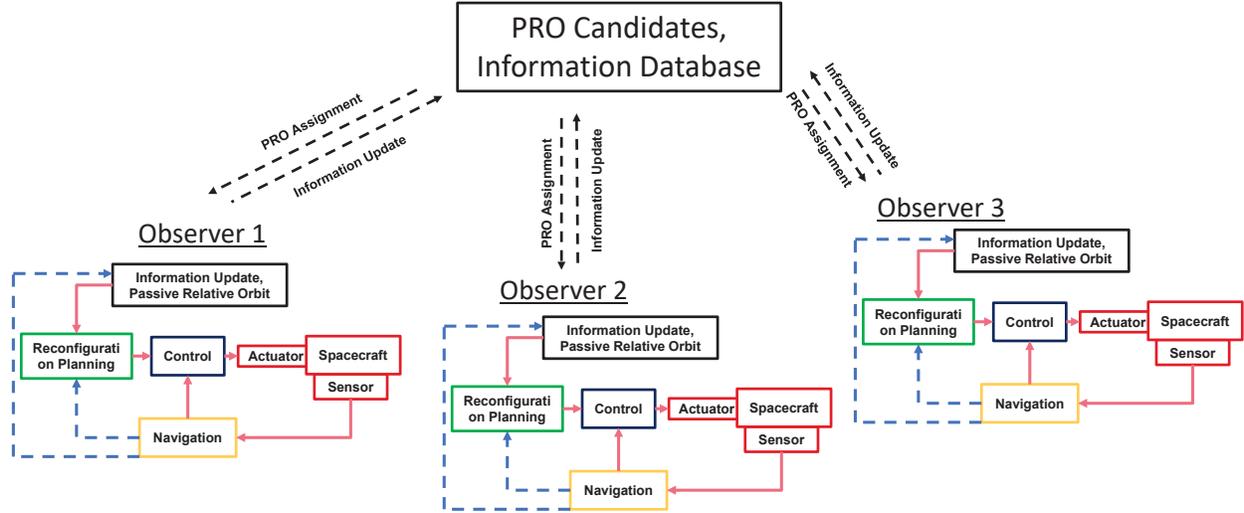
Our approach scales linearly with the number of POIs, linearly with the number of PROs, and cubically with the number of spacecraft and empirically shows very promising results with respect to the solution quality.

## G. Extension to Distributed Inspection

In the above formulation, the PRO candidate selection, the information database, information updates, and the PRO assignment play key role in distributed inspection. The distributed implementation of the architecture is shown in Fig. 9. The candidate selection can be done by one spacecraft and the resulting set of selected PROs can be communicated efficiently to the others using broadcast communication. It is also possible to compute this in a distributed fashion by using synchronization messages between the spacecraft. The database and the information update can be updated in a distributed fashion using broadcast communication, where each observer broadcasts its' current camera pose at a fixed frequency. The PRO assignment can be easily distributed by using, for example, distributed auctions [18]. Note that the PRO controller, the attitude planning, and control are decentralized, provided we have the full absolute pose estimates of the observers and the LVLH frame.

# IV. Inspection of a Target Spacecraft

In this section, we apply the proposed multi-level hierarchical GNC architecture and its algorithms towards inspection of the Cygnus spacecraft (target) in a low Earth orbit using multiple CubeSat observers. The choice of Cygnus as the target is motivated by a recent mission Seeker [4], where a single CubeSat was deployed from the target spacecraft and initialized in a relative orbit around Cygnus. The CubeSat was used to estimate the centroid of a bounding box around the target spacecraft using a neural network. With the potential applications of multi-spacecraft extension to this



**Fig. 8 Centralized PRO candidate selection and information database implementation of the framework in Fig. 3 for inspection.** The sampled PRO candidates and the information database are stored in the memory of the target spacecraft. The PRO candidates are then assigned to the observers based on coverage (information) and fuel optimality. The assigned candidates are communicated to the observers. The observers have a decentralized controller for initialization or reconfiguration to the assigned PROs. From the PRO, the observer spacecraft communicates the information update to the target spacecraft at a fixed time period for computing a new set of PRO candidates to inspect areas that have not been inspected earlier.

mission, we study the mission scenario where multiple CubeSats are deployed from the target spacecraft and initialized in safe stable relative orbits to inspect Cygnus. We begin by describing the problem setup, followed by implementation details, and finally present results for planning and control during different phases of the mission.

### A. Inspection Scenario Setup

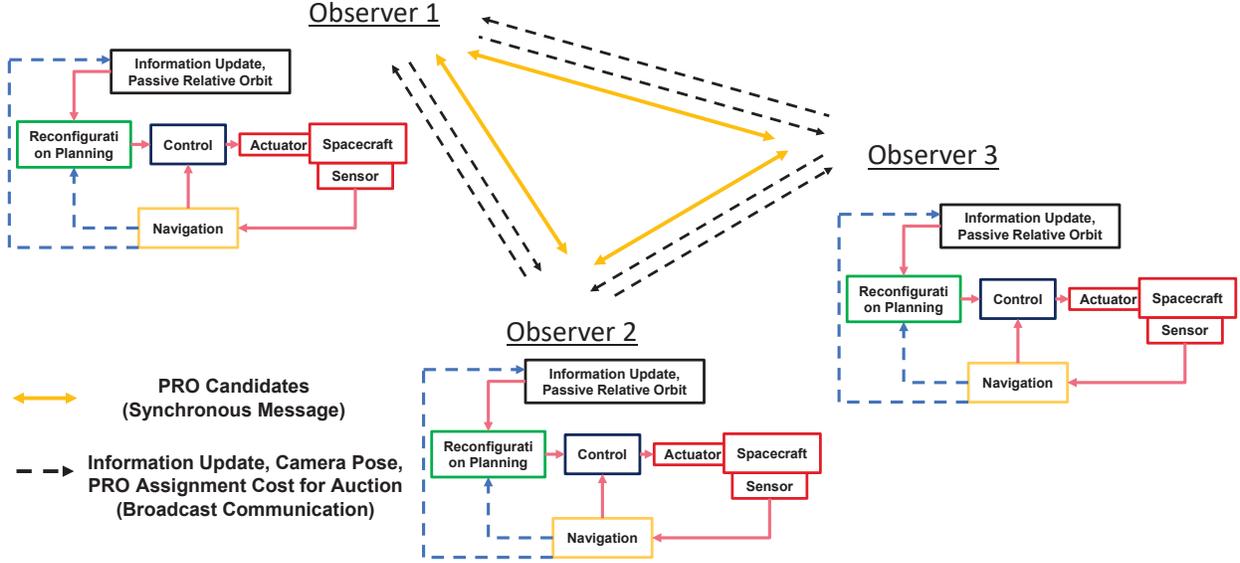
The observer spacecraft are of mass  $m = 10$  kg corresponding to a 6U-CubeSat ( $20\text{cm} \times 10\text{cm} \times 30\text{cm}$ ) and the thruster coefficient is  $f_{\text{thr}} = 200$  mN for each element of the input scaling vector  $\mathbf{u} \in [-1, 1]$  for the force per unit mass in Eq. (3). Each CubeSat is deployed using a Nanoracks [30] system with initial velocity between 0.5 m/s to 2 m/s from Cygnus. The target spacecraft is in an orbit with radius  $r_0 = 6771$  km. The number of observers deployed from Cygnus is varied from 1 to 5 to study the design trade-off for the inspection mission. We assume that the observers are equipped with a reaction wheel configuration that is capable of performing attitude control for a maximum inertia  $0.2 \text{ kgm}^2$  in each direction and have a limit maximum torque of  $\pm 2$  mNm in each direction. For each case, the proposed architecture in Algorithm 1 is applied to initialize stable orbits and to compute the progress of inspection in PROs over time. In the following test cases, we implement the proposed algorithm on a single computer (could be in the target spacecraft or in an observer spacecraft) and communicate the assigned PROs, the reconfiguration plan, and the information database to the observers.

### B. Results and Discussion

In this section, we demonstrate the effectiveness of the proposed architecture in inspecting Cygnus using multiple spacecraft through orbits and reconfiguration planning based on the information cost function  $H$ . Additionally, we discuss in detail how the proposed architecture can be utilized as a design tool to determine the optimal number of observers needed and to validate the hardware selection for multi-spacecraft inspection missions.

#### 1. Deployment and PRO Initialization

The algorithm is initialized when the CubeSats are deployed from Cygnus using a Nanorack. The initial deployment of the CubeSat plays a crucial role in the total fuel required for the mission – we design for this by using a nominal



**Fig. 9** Distributed PRO candidate selection and information database implementation of the framework in Fig. 3 for inspection. The information database is stored on all the observers and evaluated using the updates communicated at fixed frequency from the neighbours. The PRO candidates are sampled in a distributed fashion using synchronous messages with the updated information cost. Optimal PRO assignment is done using distributed auction method discussed in [18].

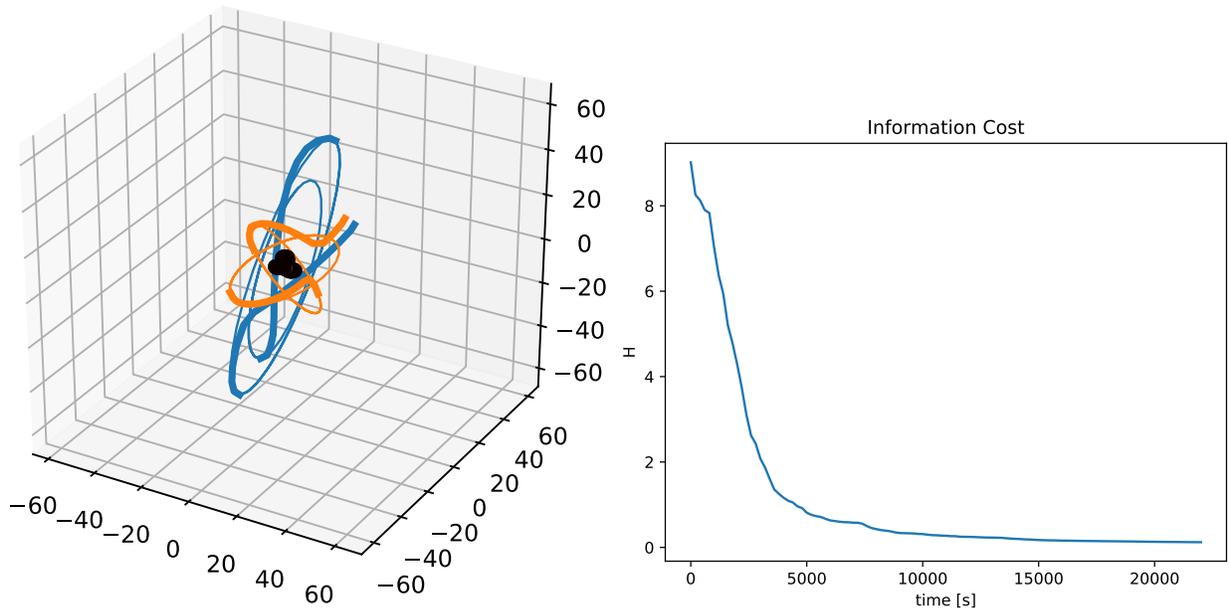
initial velocity condition that is 0.5 m/s in our algorithm to get an estimate of the fuel required to complete the mission. The initial position after drift is randomly sampled within 10 m to 20 m away from the Cygnus spacecraft. The average PRO initialization cost is computed to be 0.09 for zero initial velocity, 1.93 for the nominal deployment velocity of 0.5 m/s, and 7.91 for the worst case deployment velocity. Due to the high cost and potential violation of HCW dynamics assumption for large initial velocity deployment ranging from 0.7 m/s to 2 m/s, we recommend that the observers drift initially to get close to Cygnus over the next few orbits for minimum fuel initialization. This approach requires further analysis and is deferred to a future paper.

## 2. Inspection with Two Observers

With a nominal initial deployment position (10 m to 20 m) and velocity (0.5 m/s) knowledge, we apply Algorithm 1 with two observers ( $N = 2$ ) to inspect Cygnus. Figure 10 shows how the proposed architecture effectively coordinates the actions of the two observers to collectively inspect Cygnus over time. The two observers are initially reconfigured to a closest PRO set that minimizes  $H$  rapidly. The two observers then reconfigure twice during the entire mission to avoid information saturation. Note that we use a time-triggered scheduler to check if a reconfiguration is required to inspect areas with higher variance. The observers might choose not to reconfigure if the current orbit is already optimal. This approach can be easily modified to have an information-based trigger for quicker maneuvering response to collect measurement on POIs of higher relative importance.

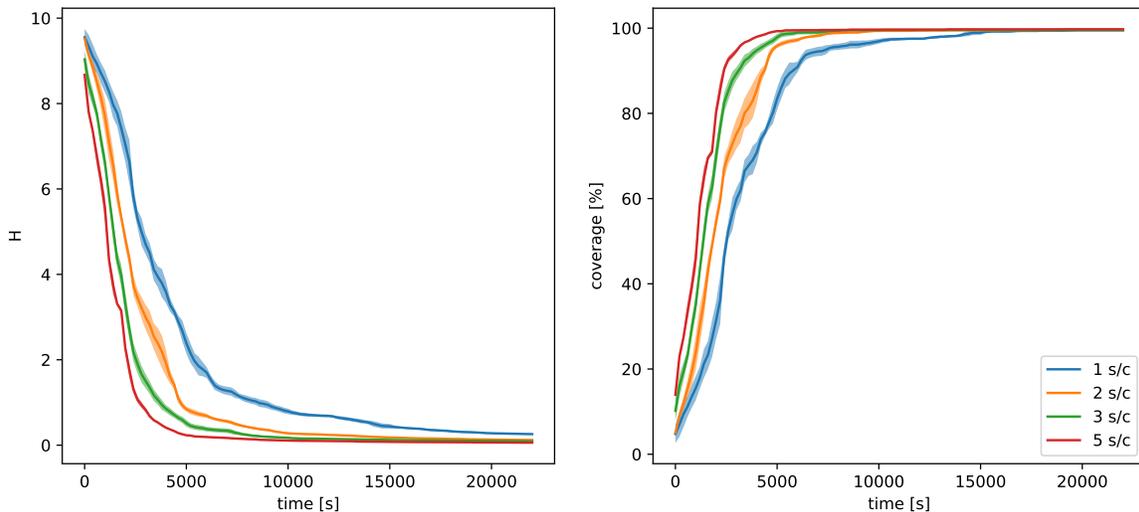
## 3. Multi-Spacecraft Formation Design For Inspection

From a mission point of view, we want to design a formation that can inspect Cygnus in minimum time and fuel efficient manner while ensuring safety. Using the PROs to generate candidate solutions already ensures minimal fuel requirement due to passive stability of orbits. We use fuel for intermediate reconfiguration between PROs, if required, which is guaranteed to be minimal from the SCP formulation in Problem 2 for reconfiguration. In order to design the optimal number of observers required for such a mission, we compare the information cost reduction by varying the formation size from 3 to 5 iteratively. The variation of information cost  $H$  and coverage over time for different formation sizes are shown in Fig. 11. From this figure, we can observe a significant time reduction in inspection when we use a formation of size 3 compared to when only using a single spacecraft. On the other hand, the coverage does not

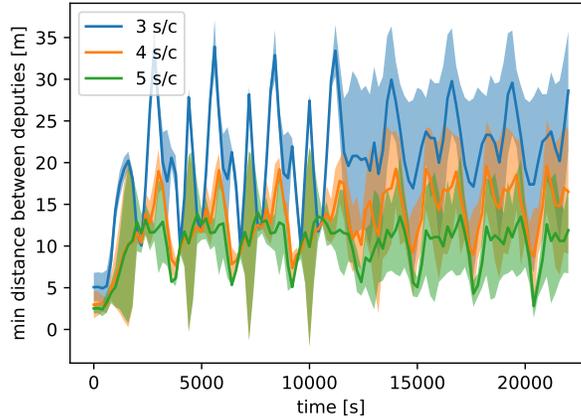


**Fig. 10** Inspection of Cygnus using two observers. Left: Trajectories of the two observers. The bold part shows the planned PRO reconfiguration. Right: The information cost converges to zero over time, because the entire target spacecraft is inspected.

increase much with time when the formation size is increased from from 3 to 5. In Fig. 12, we show the minimum distance between the observers as a function of time for different formation sizes. For the formation with 3 observers, the minimum distance varies from 5 m to 30 m as the mission evolves in time and for 5 observers, it varies from near 0 m to 15 m. There is a safety hazard when using a formation with 5 observers with no significant improvement in performance compared to formation with size 3. From this analysis, we conclude that from a minimum time and safety point of view, the mission can be achieved optimally using a formation of 3 observers.



**Fig. 11** Information cost and POI coverage over time with varying number of observer spacecraft. The results are averaged over 5 trials with standard deviation shown as shaded area. Left: Using multiple observers allows to significantly reduce the time until a target information cost value is reached. For example,  $H = 2$  is achieved in less than half the time when using 5 observers compared to when using one observer. Right: A lower information cost leads to a better coverage of POIs.



**Fig. 12** A plot showing comparison of minimum distance between observers when 3, 4, and 5 CubeSats are used for the inspection task as function of time. As expected, 3 observer configuration has larger separation compared to 5 observer configuration.

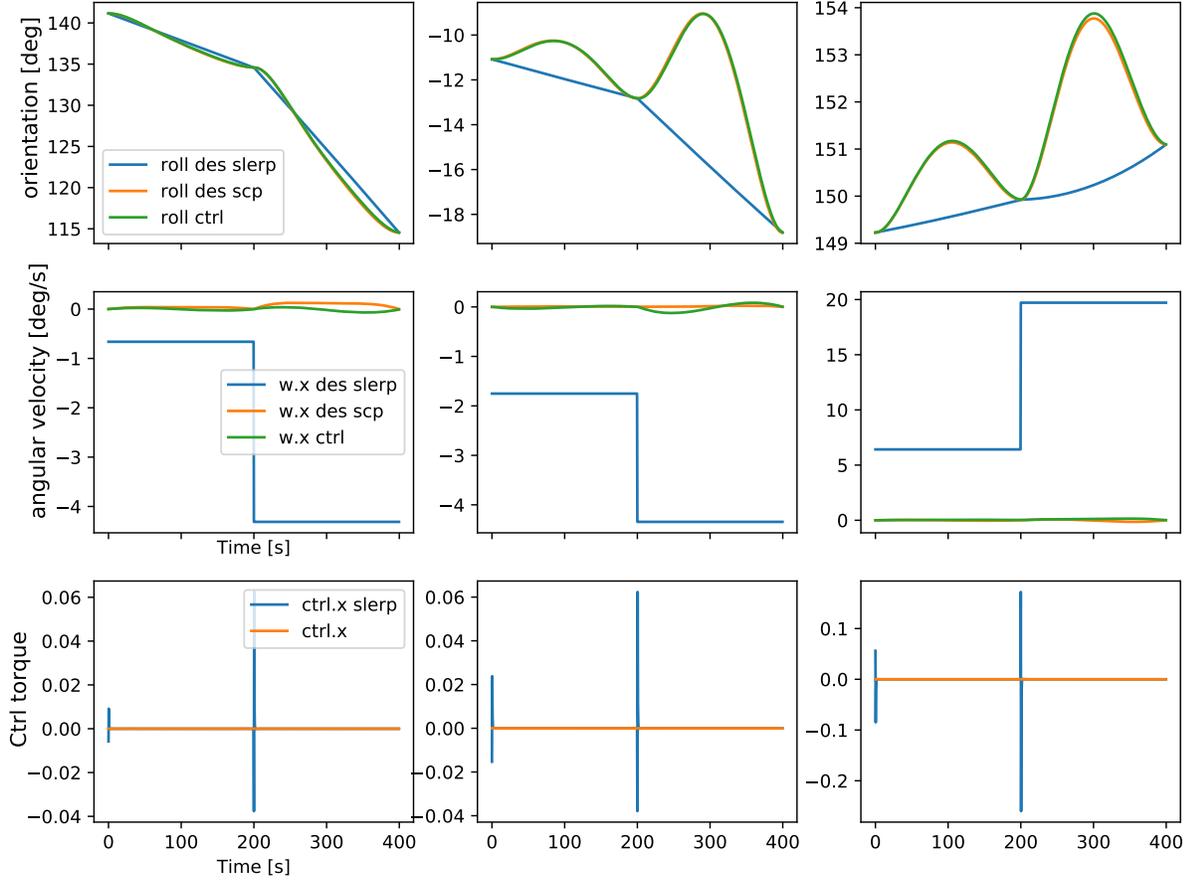
At a high-rate we run an attitude trajectory optimization and controller to smoothly track the coarse pointing vectors computed using the ray casting database. Figure 13 shows a comparison of slerp interpolated trajectory, smoothed trajectory taking into account torque and momentum limits of the reaction wheel, and the controlled trajectory. We can observe that the smoothed trajectory does not saturate the reaction wheels, whereas the slerp trajectory leads to saturation by crossing the maximum torque limit of 2 mNm and jerk in the attitude motion of the observer. Note that this step is useful in designing the pointing requirement and reaction wheel desaturation strategy for each observer. For example, the observers around Cygnus initialized in a stable PRO have their period matched to that of Cygnus’s orbital period. Due to this, the observers spend some time of the orbit in the dark and might not be able to inspect the Cygnus. The actuator limits can be used to design a partial orbit inspection strategy, where the observers only inspect when the pointing strategy is feasible for 30 % of the orbit. If the pointing plan is not feasible, then the system can desaturate the wheels when in the dark and inspect only when in the sunlight.

### C. Preliminary Hardware Experiments

In preliminary experiments, we use the information based planning architecture in Fig. 3 to optimally inspect the four surfaces of a CubeSat model as shown in Fig. 14. With known sensor model and safety constraints, the Observer-1 in Fig. 14 communicates with the neighboring spacecrafts to sort and select the unobserved surfaces of the CubeSat model. As seen in Fig. 14, there are two unobserved surfaces of the CubeSat. Observer-1 then selects a terminal state from the random samples of the feasible states around the CubeSat. The terminal state is chosen such that the observer has good visibility of the surface to be inspected from that state and requires minimum fuel to achieve that state. The terminal condition is then used as a constraint in Problem 1 for planning a safe trajectory to approach and point an observer to inspect the surfaces that are not inspected by the neighbouring observers. The planned trajectory is executed using an exponentially stable nonlinear feedback controller [31]. Implementation of the information cost on the Caltech’s M-STAR [31] testbed for computing optimal and safe terminal states of the observers is an ongoing effort to achieve real-time collaborative inspection in future.

## V. Conclusion

In this paper, we present a new guidance and control architecture that consist of multiple timescale modules for cooperatively observing and inspecting a target spacecraft in Earth orbit using multiple observer spacecraft. The proposed methodology hierarchically combines optimal trajectory design for orbit initialization and reconfiguration, attitude/pointing planning and control, and an optimal control formulation that incorporates an information cost for inspecting the target spacecraft. The information cost is used to cover the entire shape or structure of the target spacecraft and balances the trade-off between gathered data quality and fuel/energy cost. The orbit planning involves designing a safe and optimal trajectory to optimally insert each observer spacecraft into passively relative orbits around the target spacecraft for conducting the inspection task. To this end, we formulate the problem as a multi-spacecraft optimal



**Fig. 13 Attitude trajectory generated using the slerp interpolation and sequential convex programming. The control required to track the slerp trajectory and the smoothed trajectory are compared. The smoothed trajectory satisfies the torque limits of  $\pm 2$  mNm, while tracking the slerp trajectory easily saturates the reaction wheels.**

control problem and solve it using sequential convex programming. The passively stable relative orbits are chosen in an offline phase based on the energy matching condition for stability and using an information gain metric to inspect the maximum surface area of the target spacecraft. During inspection, we confirm if an area of the target spacecraft has been inspected or not, using our novel raycasting database to efficiently check the visibility of POIs. Once the passive relative orbits are initialized, we compute a pointing control plan at fast rates to optimally reorient the spacecraft for inspection. The planned attitude trajectory is executed using a nonlinear feedback controller. The effectiveness of the proposed architecture is validated via numerical simulation of inspecting the target spacecraft in Earth orbit using multiple observer spacecraft. The methods presented in this paper can be effectively used as a mission planning tool to compute and track multi-spacecraft observers on energy-efficient passive relative orbits for inspecting a spacecraft. In the future, we will consider a combined framework that integrates the relative sensing and planning that systematically accommodates for the uncertainty in estimation and thruster [32] actuation.



$$\mathbf{B}_d = \frac{f_{\text{thr}}}{m} \begin{bmatrix} \frac{1}{n^2}(1 - \cos n\Delta t) & \frac{2}{n^2}(n\Delta t - \sin n\Delta t) & 0 \\ -\frac{2}{n^2}(n\Delta t - \sin n\Delta t) & \frac{4}{n^2}(1 - \cos n\Delta t) - \frac{3}{2}\Delta t^2 & 0 \\ 0 & 0 & \frac{1}{n^2}(1 - \cos n\Delta t) \\ \frac{1}{n} \sin n\Delta t & \frac{2}{n}(1 - \cos n\Delta t) & 0 \\ -\frac{2}{n}(1 - \cos n\Delta t) & \frac{4}{n} \sin n\Delta t - 3\Delta t & 0 \\ 0 & 0 & \frac{1}{n} \sin n\Delta t \end{bmatrix} \quad (42)$$

$$\mathbf{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \quad (43)$$

### Acknowledgement

Part of this work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This work work is in part funded by the JPL-CAST Swarm Autonomy project.

### References

- [1] Long, A. M., Richards, M. G., and Hastings, D. E., "On-orbit servicing: a new value proposition for satellite design and operation," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, 2007, pp. 964–976.
- [2] Davis, J. P., Mayberry, J. P., and Penn, J. P., "On-Orbit Servicing: Inspection, Repair, Refuel, Upgrade, and Assembly of Satellites in Space," report, The Aerospace Corporation, 2019.
- [3] Piskorz, D., and Jones, K. L., "On-Orbit Assembly of Space Assets: A Path to Affordable and Adaptable Space Infrastructure," report, The Aerospace Corporation, 2018.
- [4] Banker, B., and Askew, R., "Seeker 1.0: Prototype Robotic Free Flying Inspector Mission Overview," *33rd Annual Conference on Small Satellites*, 2019.
- [5] Friend, R. B., "Orbital Express program summary and mission overview," *Sensors and Systems for Space Applications II*, Vol. 6958, edited by R. T. Howard and P. Motaghedi, International Society for Optics and Photonics, SPIE, 2008, pp. 11 – 21.
- [6] Foust, R., Chung, S.-J., and Hadaegh, F., "Autonomous In-Orbit Satellite Assembly from a Modular Heterogeneous Swarm using Sequential Convex Programming," *AIAA/AAS Astrodynamics Specialist Conference*, 2016.
- [7] Rahmani, A., Bandyopadhyay, S., Rossi, F., Wolf, M., de la Croix, J.-P., and Vander Hook, J., "Swarm of Space Vehicles and Future Opportunities," report, NASA Jet Propulsion Lab, 2018.
- [8] Solovey, K., and Halperin, D., "On the hardness of unlabeled multi-robot motion planning," *The International Journal of Robotics Research*, Vol. 35, No. 14, 2016, pp. 1750–1759.
- [9] Johnson, J. K., "On the relationship between dynamics and complexity in multi-agent collision avoidance," *Autonomous Robots*, Vol. 42, No. 7, 2018, pp. 1389–1404.
- [10] Yost, B., Agasid, E., Burton, R., Carlino, R., Defouw, G., Perez, A. D., Karacalıoğlu, A. D., Klamm, B., Rademacher, A., Schalkwyck, J., Shimmin, R., Tilles, J., and Weston, S., "Small Spacecraft Technology State of the Art," report, NASA Ames Research Center, 2018.
- [11] Nanjangud, A., Blacker, P. C., Bandyopadhyay, S., and Gao, Y., "Robotics and AI-enabled on-orbit operations with future generation of small satellites," *Proceedings of the IEEE*, Vol. 106, No. 3, 2018, pp. 429–439.
- [12] Hadaegh, F. Y., Chung, S.-J., and Manohara, H. M., "On development of 100-gram-class spacecraft for swarm applications," *IEEE Systems Journal*, Vol. 10, No. 2, 2014, pp. 673–684.

- [13] Bandyopadhyay, S., Foust, R., Subramanian, G. P., Chung, S.-J., and Hadaegh, F. Y., "Review of formation flying and constellation missions using nanosatellites," *Journal of Spacecraft and Rockets*, Vol. 53, No. 3, 2016, pp. 567–578.
- [14] Fourie, D., Tweddle, B., Ulrich, S., and Saenz Otero, A., "Vision-based relative navigation and control for autonomous spacecraft inspection of an unknown object," *AIAA guidance, navigation, and control (GNC) conference*, 2013, p. 4759.
- [15] Tweddle, B. E., and Saenz-Otero, A., "Relative computer vision-based navigation for small inspection spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 5, 2015, pp. 969–978.
- [16] Sabatini, M., Volpe, R., and Palmerini, G., "Centralized visual based navigation and control of a swarm of satellites for on-orbit servicing," *Acta Astronautica*, 2020.
- [17] Capuano, V., Kim, K., Harvard, A., and Chung, S.-J., "Monocular-based pose determination of uncooperative space objects," *Acta Astronautica*, Vol. 166, 2020, pp. 493–506.
- [18] Morgan, D., Subramanian, G. P., Chung, S.-J., and Hadaegh, F. Y., "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *The International Journal of Robotics Research*, Vol. 35, No. 10, 2016, pp. 1261–1285.
- [19] Bandyopadhyay, S., Chung, S.-J., and Hadaegh, F. Y., "Nonlinear Attitude Control of Spacecraft with a Large Captured Object," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 4, 2016, pp. 754–769.
- [20] Alfriend, K., Vadali, S. R., Gurfil, P., How, J., and Breger, L., *Spacecraft formation flying: Dynamics, control and navigation*, Vol. 2, Elsevier, 2009.
- [21] Junkins, J. L., and Schaub, H., *Analytical mechanics of space systems*, American Institute of Aeronautics and Astronautics, 2009.
- [22] Morgan, D., Chung, S.-J., Blackmore, L., Acikmese, B., Bayard, D., and Hadaegh, F. Y., "Swarm-keeping strategies for spacecraft under J2 and atmospheric drag perturbations," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, 2012, pp. 1492–1506.
- [23] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., "Model predictive control of swarms of spacecraft using sequential convex programming," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725–1740.
- [24] Foust, R., Chung, S.-J., and Hadaegh, F. Y., "Optimal Guidance and Control with Nonlinear Dynamics Using Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 4, 2020, pp. 633–644.
- [25] Markley, F. L., and Crassidis, J. L., *Fundamentals of spacecraft attitude determination and control*, Vol. 33, Springer, 2014.
- [26] Schwager, M., Rus, D., and Slotine, J. E., "Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment," *I. J. Robotics Res.*, Vol. 30, No. 3, 2011, pp. 371–383.
- [27] Schwager, M., Julian, B. J., Angermann, M., and Rus, D., "Eyes in the Sky: Decentralized Control for the Deployment of Robotic Camera Networks," *Proceedings of the IEEE*, Vol. 99, No. 9, 2011, pp. 1541–1561.
- [28] Kuhn, H. W., "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, Vol. 2, No. 1-2, 1955, pp. 83–97.
- [29] Bernhard, B., Choi, C., Rahmani, A., Chung, S.-J., and Hadaegh, F., "Coordinated Motion Planning for On-Orbit Satellite Inspection using a Swarm of Small-Spacecraft," *IEEE Aerospace Conference*, 2020, pp. 1–13.
- [30] "Nanoracks Cygnus Deployment Documentation," <https://nanoracks.com/products/external-cygnus-deployment/>, 2020. Accessed: 2020-07-06.
- [31] Nakka, Y. K., Foust, R. C., Lupu, E. S., Elliott, D. B., Crowell, I. S., Chung, S.-J., and Hadaegh, F. Y., "Six degree-of-freedom spacecraft dynamics simulator for formation control research," *AAS/AIAA Astrodynamics Specialist Conference*, 2018.
- [32] Nakka, Y. K., and Chung, S., "Trajectory Optimization for Chance-Constrained Nonlinear Stochastic Systems," *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 3811–3818.